UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,106	08/25/2003	Andrew James Booker	550-462	9839

23117 7590 03/31/2008
NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
CHEN, QING	
ART UNIT	PAPER NUMBER
2191	
MAIL DATE	DELIVERY MODE
03/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/647,106	BOOKER ET AL.	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 February 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-17,19-29,31-37 and 39-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-17,19-29,31-37 and 39-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 August 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on February 28, 2008.

2. **Claims 1-4, 6-17, 19-29, 31-37, and 39-42** are pending.

3. **Claims 1, 14, 27, and 35** have been amended.

4. **Claims 5, 18, 30, and 38** have been cancelled.

Response to Amendment

Drawings

5. The drawings are objected to because the drawings must be reasonably free from erasures and must be free from alterations, overwriting, interlineations, folds, and copy marks (see Figures 1-6). Applicant is advised to prepare the drawings using a computer drawing tool. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not

accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

6. The abstract of the disclosure is objected to because:

- The abstract exceeds 150 words in length.
- "Figure 4B" should be deleted.

Correction is required. See MPEP § 608.01(b).

Claim Objections

7. **Claims 1-4, 6-17, 19-26, and 32** are objected to because of the following informalities:

- **Claim 1** recites the limitation "said sequence of generated instructions." Applicant is advised to change this limitation to read "said corresponding sequence of generated instructions" for the purpose of providing it with proper explicit antecedent basis.
- **Claims 1 and 14** recite the limitation "the condition code." Applicant is advised to change this limitation to read "the corresponding condition code" for the purpose of providing it with proper explicit antecedent basis.
- **Claims 2-4 and 6-13** depend on Claim 1 and, therefore, suffer the same deficiencies as Claim 1.
- **Claims 15-17 and 19-26** depend on Claim 14 and, therefore, suffer the same deficiency as Claim 14.

- **Claims 7, 20, and 32** contain a typographical error: "[S]aid instruction group" should read -- said instruction *groups* --.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. **Claims 14-17, 19-29, 31-37, and 39-42** are rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure which is not enabling. The subject matter of "generating software test information" is considered to be critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976).

The claimed invention omits the critical step of generating software test information disclosed to be essential to the invention. The preambles of the independent claims (Claims 14, 27, and 35) clearly state that the claims are directed to generating software test information. However, the limitations of the claims only pertain to generating a sequence of instructions from a corresponding group of instructions. The claims do not recite any subject matter that relates generating a group of instructions with generating software test information. Applicant's specification clearly states that the present invention is directed to generating statistical information relating to the operation of software code when tested on a processor *(see Page 1: 3-*

6). Furthermore, Figure 4B and its corresponding description in Applicant's specification clearly illustrates the step of collecting profiling information.

A claim which omits subject matter disclosed to be essential to the invention as described in the specification or in other statements of record may be rejected under 35 U.S.C. 112, first paragraph, as not enabling. See MPEP § 2164.08(c). Such essential matter may include missing elements, steps, or necessary structural cooperative relationships of elements described by the Applicant as necessary to practice the invention.

10. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. **Claims 1-4, 6-17, 19-29, 31-37, and 39-42** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 14, and 27 recite the limitation "for selected instructions having a condition code the corresponding generated instruction being a predetermined generated instruction having a corresponding condition code." This is awkward claim language and thus rendered the claims indefinite. Applicant is advised to add a "wherein" clause or other means to separate the new independent clause. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "for selected instructions having a condition code, wherein a

corresponding generated instruction being a predetermined generated instruction having a corresponding condition code" for the purpose of further examination.

Claims 1, 14, and 27 recite the limitation "the corresponding generated instruction." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "a corresponding generated instruction" for the purpose of further examination.

Claims 1, 14, and 35 recite the limitation "the operation." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "an operation" for the purpose of further examination.

Claims 2-4 and 6-13 depend on Claim 1 and, therefore, suffer the same deficiencies as Claim 1.

Claims 15-17 and 19-26 depend on Claim 14 and, therefore, suffer the same deficiencies as Claim 14.

Claims 28, 29, and 31-34 depend on Claim 27 and, therefore, suffer the same deficiencies as Claim 27.

Claims 36, 37, and 39-42 depend on Claim 35 and, therefore, suffer the same deficiency as Claim 35.

Claims 9, 22, and 34 recite the limitation "that instruction group." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "an instruction group" for the purpose of further examination.

Claim 41 recites the limitation "said sequence of generated instructions." There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "a sequence of generated instructions" for the purpose of further examination.

Claims 41 and 42 recite the limitation "said step c)." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "said step b)" for the purpose of further examination.

12. **Claims 14-17, 19-29, 31-37, and 39-42** are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01. The omitted step is "generating software test information." The omitted step is considered to be critical to the claimed invention, since the step is necessary and must occur for the claimed invention to function as intended by the Applicant as described in the disclosure.

Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

14. **Claims 1-4, 6, 10-17, 19, 23-29, 31, 35-37, and 39-42** are rejected under 35 U.S.C. 102(a) as being anticipated by **Applicant Admitted Prior Art (hereinafter "AAPA")**.

As per **Claim 1**, AAPA discloses:

- a) generating, from a sequence of instructions, at least one of which includes a condition code, a corresponding sequence of generated instructions, for selected instructions having a condition code, wherein a corresponding generated instruction being a predetermined generated instruction having a corresponding condition code *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14."; Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.")*;

- b) executing, on a target processor, said corresponding sequence of generated instructions and thereby producing software test information *(see Page 6: 10-13, "The original opcode 14 together with the generated opcode 16 is stored in the memory 22. A handler routine 30 is also stored in the memory 22 which is operable by the processor core 20 to generate code coverage and profiling information using the program code 14 and the generated opcode 16.")*; and

- c) when during said step (b) said predetermined generated instruction is encountered, determining with reference to status information associated with an operation of said target processor whether the corresponding condition code of said predetermined generated instruction is satisfied and, if so, replacing said predetermined generated instruction with said corresponding instruction from said sequence of instructions so as to cause said corresponding instruction to be executed, wherein said predetermined generated instruction is an instruction which is not recognised by said target processor *(see Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16. The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction. Once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding original instruction. The handler routine 30 then determines whether the original instruction would have been executed by comparing its condition code with the current status flags of the processor core 20. If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original*

instruction. The handler routine 30 is then exited. The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.").

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein each instruction of said sequence of instructions includes a condition code *(see Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.").*

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by said target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.").*

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein said status information is predetermined architectural state associated with said target processor and said condition code specifies a status of said predetermined architectural state that must be met in order for the instruction to be executed *(see Page 2: 8-14, "Hence, by way of example, in the ARM (trademark) instruction set, the condition code EQ/NE requires that the zero condition flag (`Z` flag) must be set/cleared respectively for the instruction*

to be executed; the Z flag is set if the result of the last condition flag setting instruction was zero. Similarly, the condition code PL/MI requires that the negative condition flag (`N` flag) must be cleared/set respectively for the instruction to be executed; the N flag is set if the result of the last condition flag setting instruction was negative.").

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- generating, from said sequence of instructions, a sequence of generated instructions, a predetermined generated instruction being generated for each instruction in the sequence of instructions *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").*

As per **Claim 10**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- incrementing a coverage counter when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original*

instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").

As per **Claim 11**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- incrementing a counter associated with said corresponding instruction when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 12**, the rejection of **Claim 11** is incorporated; and AAPA further discloses:

- replacing a preceding instruction in said sequence of generated instructions with said predetermined generated instruction having a condition code corresponding to said preceding instruction *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 13**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- executing said corresponding instruction on said target processor *(see Page 6: 24 and 25, "The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.")*.

Claims 14-17, 19, and 23-26 are apparatus claims corresponding to the method claims above (Claims 1-4, 6, and 10-13) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-4, 6, and 10-13.

As per **Claim 27**, AAPA discloses:

- a) generating, from a sequence of instructions, at least one of which includes a condition code, a corresponding sequence of generated instructions, for selected instructions having a condition code, wherein a corresponding generated instruction being a predetermined generated instruction having a corresponding condition code, wherein said predetermined generated instruction is an instruction which is not recognised by a target processor *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14."; Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code."; Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16.*

The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction.").

As per **Claim 28**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- wherein each instruction of said sequence of instructions includes a condition code *(see Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.").*

As per **Claim 29**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by a target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.").*

As per **Claim 31**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- generating, from said sequence of instructions, a sequence of generated instructions, a predetermined generated instruction being generated for each instruction in the sequence of instructions *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the*

program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").

As per **Claim 35**, AAPA discloses:

- a) executing, on a target processor, a sequence of generated instructions *(see Page 6: 10-13, "The original opcode 14 together with the generated opcode 16 is stored in the memory 22. A handler routine 30 is also stored in the memory 22 which is operable by the processor core 20 to generate code coverage and profiling information using the program code 14 and the generated opcode 16.")*; and

- b) when a predetermined generated instruction is encountered during said step (a), determining with reference to status information associated with an operation of said target processor whether a condition code of that predetermined generated instruction is satisfied and, if so, replacing said predetermined generated instruction with a corresponding instruction from a sequence of instructions so as to cause said corresponding instruction to be executed, wherein said predetermined generated instruction is an instruction which is not recognised by said target processor *(see Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16. The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction. Once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding*

original instruction. The handler routine 30 then determines whether the original instruction would have been executed by comparing its condition code with the current status flags of the processor core 20. If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction. The handler routine 30 is then exited. The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.").

As per **Claim 36**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by said target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.").*

As per **Claim 37**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- wherein said status information is predetermined architectural state associated with said target processor and said condition code specifies a status of said predetermined architectural state that must be met in order for the instruction to be executed *(see Page 2: 8-14, "Hence, by way of example, in the ARM (trademark) instruction set, the condition code EQ/NE requires that the zero condition flag (`Z` flag) must be set/cleared respectively for the instruction to be executed; the Z flag is set if the result of the last condition flag setting instruction was zero.*

Similarly, the condition code PL/MI requires that the negative condition flag (`N` flag) must be cleared/set respectively for the instruction to be executed; the N flag is set if the result of the last condition flag setting instruction was negative.").

As per **Claim 39**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- incrementing a coverage counter when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 40**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- incrementing a counter associated with said corresponding instruction when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 41**, the rejection of **Claim 40** is incorporated; and AAPA further discloses:

- replacing a preceding instruction in a sequence of generated instructions with said predetermined generated instruction having a condition code corresponding to said preceding instruction *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 42**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- executing said corresponding instruction on said target processor *(see Page 6: 24 and 25, "The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.").*

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. **Claims 7-9, 20-22, and 32-34** are rejected under 35 U.S.C. 103(a) as being unpatentable over **AAPA** in view of **US 5,712,996 (hereinafter "Schepers")**.

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- a2) generating said predetermined generated instruction for one instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").*

However, AAPA does not disclose:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions.

Schepers discloses:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions *(see Abstract, "In order to be able to execute rapid processing of a program on super-scalar microprocessors, the individual instructions of this program must be divided into instruction groups, which can be processed by processing units of the microprocessor, in such a way that the instructions can be processed in parallel.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions. The modification would be obvious

because one of ordinary skill in the art would be motivated to execute rapid processing of the program code *(see Schepers – Abstract)*.

As per **Claim 8**, the rejection of **Claim 7** is incorporated; and AAPA further discloses:

- generating said predetermined generated instruction for the last instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.")*.

As per **Claim 9**, the rejection of **Claim 7** is incorporated; however, AAPA does not disclose:

- wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group.

Schepers discloses:

- wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group *(see Column 2: 63-65, "The number of the components per instruction group is fixed in accordance with the number of the instructions which a microprocessor can load simultaneously.")*.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group. The modification would be obvious because one of ordinary skill in the art would be motivated to determine the number of instructions a microprocessor can load simultaneously *(see Schepers – Column 2: 63-65)*.

Claim 20 is rejected for the same reason set forth in the rejection of Claim 7.

Claim 21 is rejected for the same reason set forth in the rejection of Claim 8.

Claim 22 is rejected for the same reason set forth in the rejection of Claim 9.

As per **Claim 32**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- a2) generating said predetermined generated instruction for one instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.")*.

However, AAPA does not disclose:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions.

Schepers discloses:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions *(see Abstract, "In order to be able to execute rapid processing of a program on super-scalar microprocessors, the individual instructions of this program must be divided into instruction groups, which can be processed by processing units of the microprocessor, in such a way that the instructions can be processed in parallel.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions. The modification would be obvious because one of ordinary skill in the art would be motivated to execute rapid processing of the program code *(see Schepers – Abstract)*.

As per **Claim 33**, the rejection of **Claim 32** is incorporated; and AAPA further discloses:

- generating said predetermined generated instruction for the last instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the*

instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").

As per **Claim 34**, the rejection of **Claim 32** is incorporated; however, AAPA does not disclose:

- wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group.

Schepers discloses:

- wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group *(see Column 2: 63-65, "The number of the components per instruction group is fixed in accordance with the number of the instructions which a microprocessor can load simultaneously.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include wherein said predetermined generated instruction provides information relating to the number of instructions in an instruction group. The modification would be obvious because one of ordinary skill in the art would be motivated to determine the number of instructions a microprocessor can load simultaneously *(see Schepers – Column 2: 63-65).*

Response to Arguments

17. Applicant's arguments with respect to Claims 1, 14, 27, and 35 have been considered, but are moot in view of the new ground(s) of rejection.

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/QC/
March 25, 2008
/Wei Zhen/
Supervisory Patent Examiner, Art Unit 2191